Amendments to the Specification

1. Please replace paragraph [0013] with the following amended paragraph:

[0013] As seen with respect to Figure 1, a system 10 for managing power of digital circuitry includes a power intensive software application module 12 that generates data, temporarily storing the generated data in a data buffer 14. The data buffer 14 feeds data to software application module18. Software application modules 12 and 18 can be independent applications that pass data to each other through data buffer 14, or can be interacting components of a single software application that use the data buffer 14 for optimizing processing efficiency and throughput. In accordance with the present invention, data buffer 14 levels can be monitored by a processor clock speed or voltage select 16.

A1

---

2. Please replace paragraph [0014] with the following amended paragaraph:

[0014] In operation, a module 12 acts as a data source, generating data at a variable rate for transfer to a data buffer 14. At a given processor frequency, this rate variation can be highly variable, and is mostly a function of the actual data being processed by the software application (which includes, but is not limited to modules 12 and 18), and the processor load from other applications. The module 18 acts as a sink, processing data at a fixed, or slowly varying rate. For the overall application to work

A2

properly, there should always be enough data in the buffer to sustain that rate. The rate can be modulated by changing the processor frequency. In effect, changes in the data buffer 14 level act to control the processor voltage and frequency, with the voltage and frequency increasing when the buffer level is small. Conversely, voltage and frequency are decreased when the buffer level is high. If the software application can directly control the processor voltage and frequency, the buffer level can be used to directly control the processor states. This may be done indirectly as well, by interfacing to a performance-control application that directly controls the processor. Otherwise, the value of the buffer level can be used as the information (directly or indirectly) passed to an operating system or a hardware power management system.

A2

---

3. Please replace paragraph [0015] with the following amended paragraph:

[0015] This method can also be used for those applications for which the module 12 has to forward data to the data buffer 14 at predetermined constant rates (or a slowly varying ~~ratebound~~ rate bound by a known value). As before, variation of the data rate in the module 18 at a given level of processor performance is mostly a function of the actual data being processed by the application, and the processor load from other applications. To prevent buffer overflow, the level of the data buffer 14 indirectly controls the frequency and voltage of the processor by increasing voltage & frequency when the buffer level is high, and reducing voltage and frequency when the buffer level is low. As will be appreciated, certain applications may have constraints on both incoming and

A3

outgoing rates from the data buffer 14. A combination of the policies described above can then be used.

---

4. Please replace paragraph [0016] with the following amended paragraph:

[0016] The foregoing method allows dynamic power management of a digital circuitry, including conventional processors, graphic processors, or processors optimized for network or portable applications. Typically, targeted applications are power intensive media or audiovisual encoding, decoding, or other data manipulation that consumes and/or generates substantial amounts of data that is storable in a buffered pipeline. The frequency/voltage of digital circuitry and transition times between various frequencies or voltages are controlled by monitoring the level of the appropriate data buffers, which either already exist to support the application or can be explicitly added.

---

5. Please replace paragraph [0017] with the following amended paragraph:

[0017] As seen with respect to Figure 2, a control scheme 20 illustrates switching back-and-forth between a high power state and a low power state in accordance with two state transition diagram 24. The state transition diagram 24 is intended for processor performance control while operating in buffer underflow conditions, with the data buffer being fed a highly variable data stream. The data buffer is defined to have a minimum lower level B_0 and higher level B_1 (schematically illustrated by buffer block ~~32~~ 22). The time variable level B controls the frequency of a processor with two voltage/frequency

states (state 1 is lower frequency than state 2). Assuming that the processor is initially in state 1, it is switched to state 2 if the buffer level becomes too small (smaller than B_0). It switches back to state 1 when the buffer level becomes larger than B_1.

A5

---

6. Please replace paragraph [0019] with the following amended paragraph:

[0019] For clarity, the variables B_0, B_1, B_2 and B_3 in both Figures 2 and 3 are represented as being static. The overall buffer size of the buffer block 32 is similarly fixed. As will be appreciated, however, the buffer parameters can be modified as a function of other application parameters. For example, in a compressed video playback application, the variables could be changed as a function of the media content, bit rate, type of encoding (constant versus variable bit stream), sequence structure (number of I/P/B frames), etc. Adaption during the playback of a sequence as a function of compressed frame size (instantaneous bit rate), frame types, and read-access may similarly assist in power optimization. For example, before a load operation from the media storage (e.g., hard-drive, CD-ROM), it can be useful to guarantee a larger minimum number of frames in the buffer because of the increased system load. For applications with very low latency, the values of B_0, B_1, B_2 and B_3 could also be made time dependent.

A6
